

Тест-требования

Цель: Проверить корректность работы метода CalculateFuelConsumption, который рассчитывает расход топлива для заданного пути между двумя точками на карте.

Требования:

1. Корректность расчета расхода топлива:

- Метод должен корректно рассчитывать расход топлива для существующего пути между двумя точками.
- Ожидаемый результат должен соответствовать формуле.

2. Обработка отсутствующих путей:

- Метод должен корректно обрабатывать ситуацию, когда путь между точками не существует (расстояние равно `double.PositiveInfinity`).
- В этом случае метод должен возвращать `double.PositiveInfinity`.

3. Проверка граничных значений:

- Метод должен корректно работать с минимальными и максимальными значениями расстояний и расхода топлива.

4. Проверка некорректных входных данных:

- Метод должен корректно обрабатывать ситуации, когда входные данные некорректны (например, отрицательные значения расстояний или расхода топлива).

Тест-план

Объект тестирования: Метод CalculateFuelConsumption.

Область тестирования:

- Позитивные тесты: проверка корректности расчета расхода топлива.
- Негативные тесты: проверка обработки отсутствующих путей и некорректных входных данных.

Тестовые сценарии:

1. Позитивный тест: Корректность расчета расхода топлива

- **Описание:** Проверить, что метод корректно рассчитывает расход топлива для существующего пути.
- **Входные данные:**

- Матрица расстояний: {{0, 0.94, double.PositiveInfinity}, {0.94, 0, 0.66}, {double.PositiveInfinity, 0.66, 0}}
- Расход топлива: 10.0 литров на 100 км
- Начальная точка: 0
- Конечная точка: 1
- **Ожидаемый результат:** 0.094 литра

2. Негативный тест: Обработка отсутствующих путей

- **Описание:** Проверить, что метод возвращает double.PositiveInfinity, если путь между точками не существует.
- **Входные данные:**
- Матрица расстояний: {{0, double.PositiveInfinity, double.PositiveInfinity}, {double.PositiveInfinity, 0, 0.66}, {double.PositiveInfinity, 0.66, 0}}
- Расход топлива: 10.0 литров на 100 км
- Начальная точка: 0
- Конечная точка: 1
- **Ожидаемый результат:** double.PositiveInfinity

3. Тест граничных значений: Минимальные и максимальные значения

- **Описание:** Проверить, что метод корректно работает с минимальными и максимальными значениями расстояний и расхода топлива.
- **Входные данные:**
- Матрица расстояний: {{0, 0.001, double.PositiveInfinity}, {0.001, 0, 1000}, {double.PositiveInfinity, 1000, 0}}
- Расход топлива: 0.1 и 1000.0 литров на 100 км
- Начальная точка: 0
- Конечная точка: 1
- **Ожидаемый результат:** Корректный расчет расхода топлива для заданных значений.

4. Негативный тест: Обработка некорректных входных данных

- **Описание:** Проверить, что метод корректно обрабатывает отрицательные значения расстояний или расхода топлива.
- **Входные данные:**
- Матрица расстояний: {{0, -0.94, double.PositiveInfinity}, {-0.94, 0, 0.66}, {double.PositiveInfinity, 0.66, 0}}
- Расход топлива: -10.0 литров на 100 км
- Начальная точка: 0

- начальная точка: 0

- Конечная точка: 1

- **Ожидаемый результат:** Метод должен выбросить исключение или вернуть ошибку.

Критерии завершения тестирования:

- Все тестовые сценарии успешно выполнены.
- Ошибки исправлены, и повторное тестирование прошло успешно.

Ресурсы:

- Разработчики и тестировщики, участвующие в проекте.
- Среда выполнения тестов (например, Visual Studio, GitHub Actions).